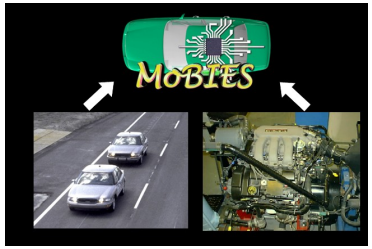
Model-Based Integration Of Embedded Software

PI Meeting

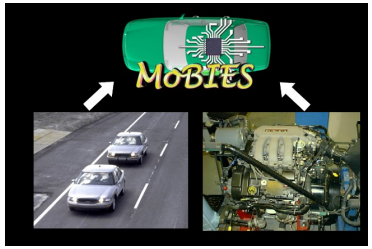
July 24 – 26, 2002
New York, NY



Smart Vehicles: An Open Testbed for Design, Testing and Implementation for Automotive Embedded Systems

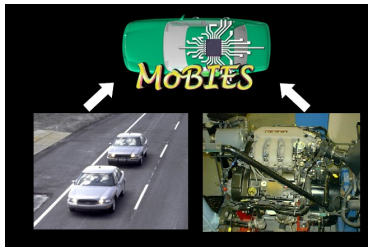
Pravin Varaiya, (510) 642-5270, varaiya@eecs.berkeley.edu
J. Karl Hedrick, (510) 642-2482, khedrick@newton.berkeley.edu
The University of California, Berkeley

Subcontractors and Collaborators



- **Subcontractor: Teja**
  - **Goal: Provision of baseline hybrid systems modeling environment**
- **Collaborators: Ford, General Motors, Motorola**
  - **Goals: Automotive industry user perspective and inputs, particularly with technology evaluation and transition**
- **Collaborative Effort with Software Enabled Control Program:**
  - **Ptolemy, Giotto**

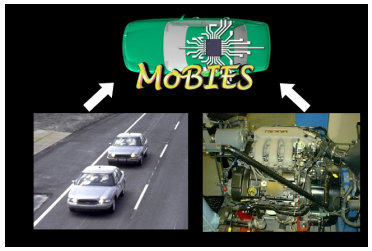
Problem Description and Program Objective



- **Problem Statement**

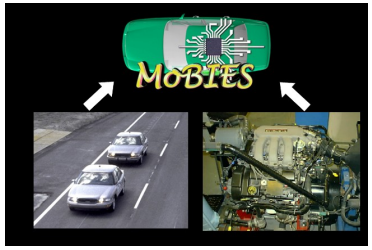
  "Auto OEP" lead in developing customizable software framework for model based system integration, with an aim to significantly improve embedded system design process
- **SmartVehicle Contribution**
  - **Provision of vehicle and engine domain-specific libraries for application development**
  - **Definition and execution of high impact physical challenge problems**
    - Advanced Vehicle Control Testbed: multi-agent sensing and control ("Cooperative Adaptive Cruise Control")
    - Engine Control System Rapid-Prototyping Testbed: powertrain control for emissions
  - **Co-develop performance measures (with auto industry partners) and provide benchmark frameworks to assess Phase I contributions**
  - **Foster technology transfer**

Problem Description and Program Objective (Cont'd)



- **Success Criteria**
  - **Implementation of Phase I embedded systems framework in SmartVehicles open platform for passenger vehicles**
  - **Technology transfer to vehicle engineering community**
    - Reduced development time
    - Ease in which domain experts and software engineers can interact
    - Ease in which different domain experts can specify and design code independently of one another
    - Usability, i.e., the 'naturalness' of the modeling language from which code is generated
    - Ease with which test engineers can modify and tune code in the field
    - Degree of accurate automated documentation in design

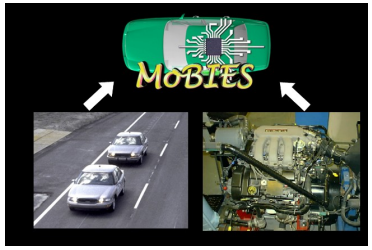
Project Status



- **Current Technical Approach**
  - **Evaluation of end-to-end tool chains**
  - **Further development and near-completion of baseline tool chains**
    - Vehicle-to-vehicle
    - Powertrain
  - **Enhancement of OEP experiments**
    - Vehicle-to-vehicle
    - Powertrain
- **Change Since Last PI Meeting**

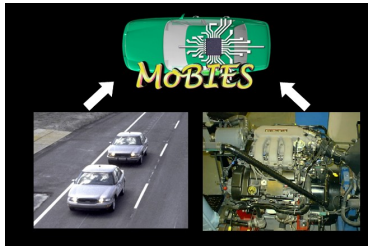
  Refinement and expansion of OEP experiments

Project Status (Cont'd)



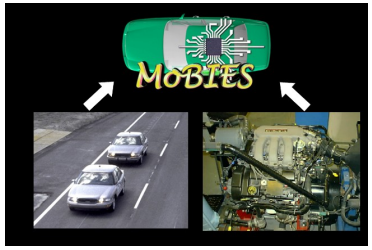
- **Deliverables and Publications**
  - **Deliverables**
    - Initial iterations of Phase I evaluations
    - Refined ETC/powertrain and V2V models
  - **Publications**
    - ETC and V2V model documentation updates
      - With updates
    - Conferences:
      - ACC, MED2002, CDC2002, etc

Project Status (Cont'd)



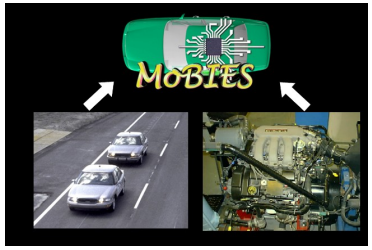
- **Specific Milestones Accomplished**
  - **Demonstration of baseline MoBIES approach**
  - **Refinement of midterm experiments**
  - **Continued coordination and cooperation with**
    - Phase I participants
    - Auto industry

Project Status (Cont'd)



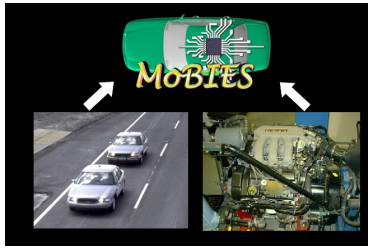
- **Midterm Experiments**
  - **Goals and success criteria (contributed from auto industry partners)**
    - **Tool chain integration**
    - **Completeness of design suite**
    - **Time and effort to configure and learn tools**
    - **Domain reconfigurability /meta-programmability**
    - **Model and code validation and verification**
    - **Relevance to automotive platforms/architecture**
    - **Integration with automotive tools**

Project Status (Cont'd)



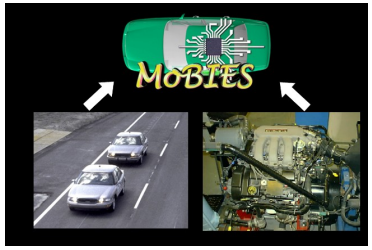
- **Midterm Experiments (Cont'd)**
  - **Other goals and success criteria**
  - **Ease of implementation**
    - Does the generated code compile and/or run? (y/n)
    - How long does it take to get the controller to run on the car?
    - How much effort must be put into interfacing with the rest of the system (database, other control code, communications system, hardware etc…)?
  - **Controller performance**
    - Robustness, Stability (proofs?)
    - Safety
    - "Drivability", that is quality of ride (comfort, etc…)
    - Was the controller's behavior accurately predicted in simulation?
  - **Real-time performance and timing aspects**
    - Is the code schedulable?
    - Are we making optimal use of the system resources?
  - **User-friendly aspects**
    - Is the code readable? Documented?
    - How long does it

Project Plans



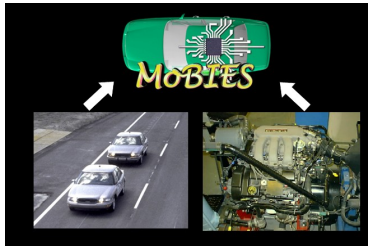
- **Next 6 Months**
  - **Conduct enhanced OEP experiments**
    - Focus on performance vis-à-vis baseline
  - **Continue to assist auto industry in evaluation**
- **Specific Performance Goals**
  - **Successful integration experiments**
  - **Baseline tool chains executed**
  - **Performance assessment**

Project Plans (Cont'd)



- **Schedule and Goal for Midterm Experiments**
  - **Powertrain**
    - See detailed presentation
  - **V2V**
    - See detailed presentation

Technology Transfer and Program Issues



- **Technology Transfer via *SmartVehicle Technology Advisory Committee***
  - **Auto OEP Partners: Ford, General Motors and Motorola**
- **Program Issues**
  - **UC Berkeley's stepped-up role in OEP integration (carried forward from last PI meeting)**
  - **Process of transitioning technology to automotive industry (carried forward from last PI meeting)**
    - We observe that limited resources are available from Auto OEP participants for this
    - We expect to solicit additional auto industry input for V2V experiment